

---

**Navigation and Ancillary Information Facility**

# **Modernization of SPICE Toolkits**

**June 2015**

**SPICE components and services are not restricted under ITAR and EAR regulations**



# Topics

---

Navigation and Ancillary Information Facility

- **Overview of current SPICE Toolkits**
- **An outline of plans for “Modernization of SPICE”**
- **Backup: details about current Toolkits**



---

**Navigation and Ancillary Information Facility**

# **Current Status**



# Current SPICE Toolkit Products

---

Navigation and Ancillary Information Facility

- **The SPICE Toolkit is available in Fortran, C, IDL (Interactive Data Language), and Matlab.**
- **The Fortran, C, IDL, and Matlab Toolkits are packaged and delivered as standalone products.**
  - The IDL and Matlab Toolkits, by necessity, also include the complete C Toolkit.
- **An “alpha-test” version of a Java Native Interface (JNI) Toolkit has also been available for several years.**
  - Has seen fairly widespread, successful use.
  - Other priorities have so far prevented NAIF from completing this product.
- **NAIF has also assembled much of a Python Toolkit, but not enough to offer it to others.**
  - Several SPICE users have built their own, probably partial Python interfaces to CSPICE and are offering these to interested persons.
  - (NAIF staff have not tried using these.)

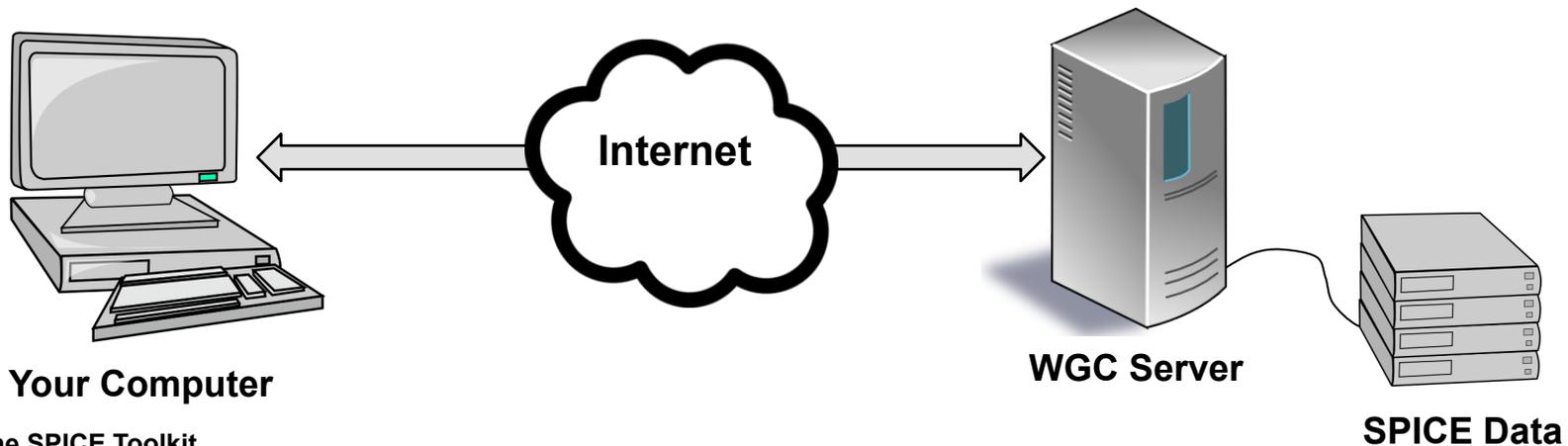




# WebGeocalc (WGC)

Navigation and Ancillary Information Facility

- **Recently NAIF made available a GUI interface to a SPICE geometry engine, using a client-server architecture**
  - <http://naif.jpl.nasa.gov/naif/webgeocalc.html>
  - The WGC user only needs a computer running a web browser.
    - » The user enters directives using typical GUI widgets.
  - The browser connects via Internet to a WGC geometry engine running on a server, such as at NAIF.
    - » Numeric answers and optional plots are displayed in your browser window; can also be downloaded.





# Toolkit Contents

---

## Navigation and Ancillary Information Facility

- **Software**
  - Subroutine libraries, with source code
    - » SPICELIB (Fortran)
    - » CSPICE (C)
    - » Icy (C)
    - » Mice (C and Matlab scripts)
  - Executable programs
    - » application and utility programs
    - » a few example programs (called “cookbook”)
  - Installation/build scripts
- **Documentation**
  - Available in ASCII and HTML
- **Example data**
  - Sample kernel files (supplied only for use with cookbook example programs, not valid for general use).



# Toolkit Characteristics

---

## Navigation and Ancillary Information Facility

- **Computations are, practically speaking, identical in all languages.**
- **For a given computer and operating system, all Toolkits use identical kernel files.**
  - Refer to the “Porting Kernels” tutorial for information about using kernels received from a machine different from what you are using.
- **Code is well tested before being released to users.**
- **New Toolkits are always backwards compatible.**
  - An application that worked when linked against an older Toolkit will link and work, without need for changes, using a new Toolkit.
  - Past functionality is never changed or removed.
    - » Enhancements of existing routines are allowed.
    - » NAIF does reserve the right to fix bugs.
- **Extensive user-oriented documentation is provided.**
  - Includes highly documented source code.



# Toolkit Versions

---

Navigation and Ancillary Information Facility

- **Toolkit Version**
  - **SPICE Toolkits have an associated Version number**
    - » **Example: “N0065” (also written as “N65”)**
  - **The version number applies to the Fortran, C, IDL and Matlab implementations for all supported platforms.**
  
- **When does NAIF release a new SPICE Toolkit version?**
  - » **Not according to a fixed schedule**
  - » **Primarily driven by availability of significant new capabilities**
    - **For example, the geometry finder subsystem**
  - » **On rare occasions a Toolkit update is released to fix bugs, improve documentation, or satisfy an urgent request from a flight project.**

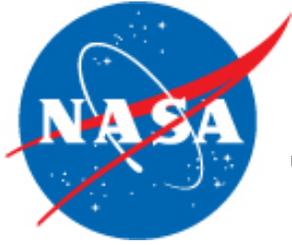


# Details on Current Toolkits

---

Navigation and Ancillary Information Facility

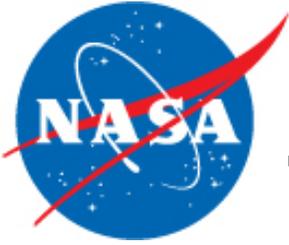
- **You may see details about the current SPICE Toolkits here:**  
[http://naif.jpl.nasa.gov/pub/naif/toolkit\\_docs/Tutorials/pdf/individual\\_docs/08\\_intro\\_to\\_toolkit.pdf](http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/08_intro_to_toolkit.pdf)



---

Navigation and Ancillary Information Facility

# Plans for SPICE Toolkit Modernization



# Toolkit Modernization

---

Navigation and Ancillary Information Facility

- **In the next few charts we outline NAIF's plans for "modernization" of the SPICE Toolkit product line.**
  - Many details are "to be determined" (TBD).
  - Schedule is also TBD



# Driving Requirements

---

Navigation and Ancillary Information Facility

- **Provide a new Toolkit environment offering object-oriented and thread-safe attributes.**
  - Details are TBD
    - » **We're particularly interested in your suggestions in this regard!**
  - To be offered on as many of the currently supported platform/OS combinations as is reasonably possible.
- **Maintain the currently offered Toolkit environments.**
  - Initially, by not changing anything.
  - Eventually, if we find we can provide the same Fortran, C, IDL and MATLAB interfaces currently available, but now connected to the underlying “modernized” Toolkit, we might be able to phase out some or all of the current Toolkits.



# Additional Objectives

---

## Navigation and Ancillary Information Facility

- **Beyond the critical requirements outlined on the previous page...**
  - Increase Toolkit functionality, such as providing higher precision time computations.
  - Reduce the volume and complexity of the code base NAIF must maintain.
  - Eliminate NAIF's dependence on the *f2c* code conversion tool.
  - Move towards a programming environment likely more familiar to people who might eventually join the NAIF Team.
  - Obtain faster execution speed where possible.



# New Toolkit Environments

---

Navigation and Ancillary Information Facility

- **We envision adding a new Toolkit environment.**
  - Will be object-oriented.
  - Will be thread-safe to some TBD extent.
  - Should provide somewhat higher performance (speed).
  - Will be provided on as many of the currently supported platform/operating system combinations as possible given the modest resources available.
    - » Exactly which ones is TBD.
  - Most functional capabilities of the current Toolkit will be maintained.
    - » Some of those capabilities will be realized within the new language, thus eliminating the need for some current APIs.
    - » We are considering NOT providing some little used current capabilities such as:
      - Events Kernel
      - Type 1 star catalog
      - Maybe a bit more... TBD



# Approach

---

## Navigation and Ancillary Information Facility

- **We tentatively plan to select C++ as the new language.**
  - **Work with some local C++ coders (nav and others) to architect the system.**
    - » **Use some existing C++ SPICE code as a testbed**
  - **Build enough C++ code to treat as an alpha-test vehicle for trial by a few interested SPICE users.**
    - » **Build Fortran and/or C wrappers for the C++ code, so we can use our existing, extensive test harness.**
    - » **Far from full functionality.**
    - » **Probably minimal documentation.**
  - **Make “corrections” as needed, then continue onward.**



# Approach, continued

---

## Navigation and Ancillary Information Facility

- **Offer the Fortran and C wrappers for the C++ code as new Toolkit environments.**
  - Would not provide object-oriented or thread safe functionality.
  - Would probably provide higher performance than the current Fortran and C Toolkits.
- **The current Icy (IDL) and Mice (MATLAB) interfaces would continue to be supported by interfacing with these new C wrappers.**
- **If the above is successful, consider phasing out the current Fortran and C Toolkits.**
  - Would need to understand any consequences and gain concurrence from end users.



# Consequences

---

## Navigation and Ancillary Information Facility

- **Undertaking this addition would reduce the amount of new and improved capabilities that would otherwise be added to the current Toolkits.**

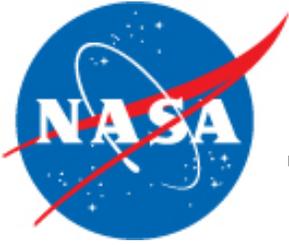


# Timeline

---

## Navigation and Ancillary Information Facility

- **“Guestimating” how long this could take is difficult.**
- **Some funding dedicated to this work is already in hand.**
  - Spread out over three years.
  - We can presumably use other SPICE funding to fill the gap.
- **A wild guess: it seems likely that release of an official C++ Toolkit would take on the order of two years starting from October 2015.**



# Your Thoughts?

---

Navigation and Ancillary Information Facility

- Do you support this endeavor?
- Have you any concerns or warnings?
- If so, please write to [charles.acton \(at\) jpl.nasa.gov](mailto:charles.acton@jpl.nasa.gov)